



11-16-2006





Opening Discussion

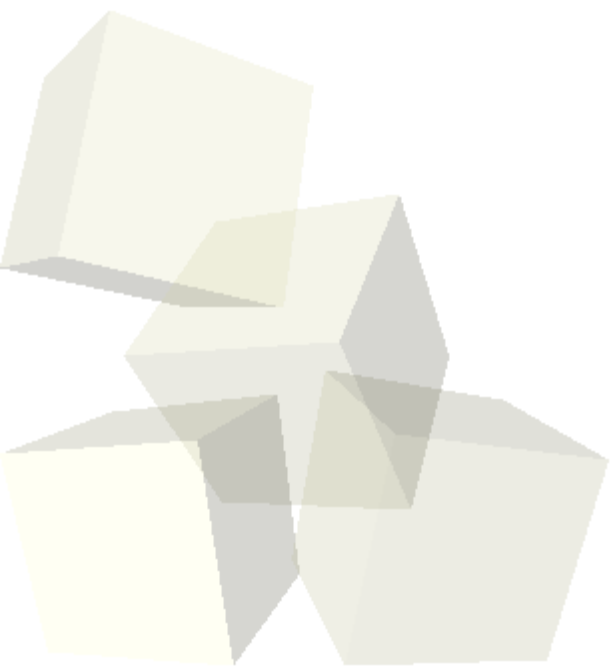
- What is a union? What are they intended to allow you to do in a program? What are the issues with unions?
- What should you be thinking of doing to optimize certain recursive functions?
- Do you have any questions about the assignment?





Optimizing Recursion

- We started making a change to our maze solver that would be faster than our original solution. Let's finish that off today.
- What are some other types of pruning that we could do for optimization?



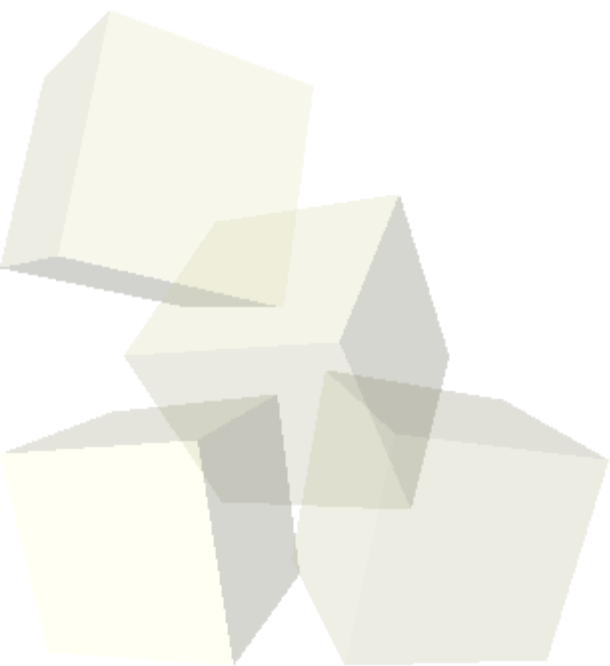


Recapping Text Files

- All the files we have talked about so far have been text files.
- Text files have a number of advantages.
 - ◆ Programatically, they are fairly easy to deal with. `fprintf` and `fscanf` are very much like `printf` and `scanf`.
 - ◆ You can edit a text file with any text editor and look at them with things like `more` and `less`.
- Text files also have some significant disadvantages.
 - ◆ You read straight through them and can't jump around.
 - ◆ Data is not at all compressed. This is why zipping a text file can shrink it so much.
 - ◆ Reading large text files is very slow. A lot of work must be done to convert from the text format to what is in memory.



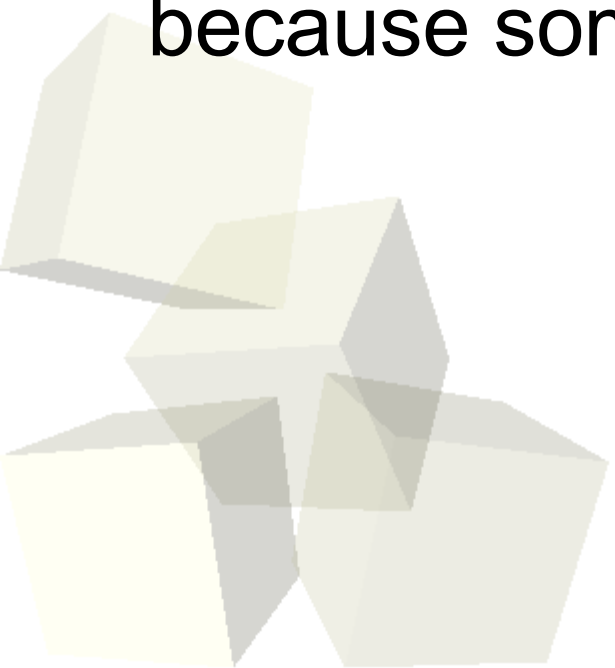
- The way to get around the problems with text files is to use binary files. Of course, this means you lose all the advantages of text files as well.
- It turns out that most programs use binary files. If you have ever looked at a Word document with a text editor or something that shows you what it really contains, you know that it is not flat text.





Opening for Binary Access

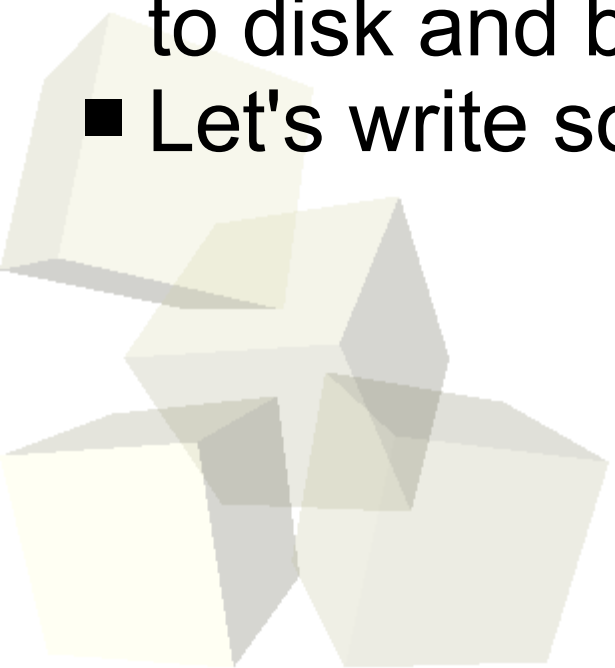
- If you want to use a binary file, all you have to do is put a 'b' in the mode string. So instead of using “wt” we would use “wb” if we want to make a new binary file to write to.
- All the other issues with fopen stay the same.
- Actually, you can do some binary access stuff if you open the file the other way, but you shouldn't because some functionality will be limited.





fread and fwrite

- After you have opened your binary file you need to read from it or write to it.
- These operations are going to be done with fread and fwrite. Look at the man pages to see what these calls look like.
- The beauty of fread and fwrite is that they basically move information straight from memory to disk and back. This makes them very fast.
- Let's write some code that uses this.





- The biggest advantage of binary files is not that they are fast and efficient. Those are very significant, but even better is that you can jump around in them randomly.
- This direct access ability lets you move to any point in a binary file and start reading or writing at that point. This is great because it allows you to treat a binary file almost just like an array.
- The binary file will be slower than a real array, but it can be much larger and it will be much faster than completely rewriting text files.
- Let's look at this idea a bit closer for a file with fixed length records.



- The way we jump around in files is with the `fseek` command.
- Let's go look at the man pages to see how this command works.
- Now we want to write some code that uses this command.





Variable Length Records

- We aren't going to code these, but binary files don't have to be made out of lots of pieces of the same size. You can use different sized pieces if you spend a little space to store an index of where each piece goes.





Pointers in Binary Files

- How do structures work in binary files?
- What happens if the structure contains a pointer?





- What types of applications might really benefit from binary files?
- Remember to turn in assignment 7/8 today.

